SXT

INSTRUCTION TEST

MD-11-DCKBA-B

EP-DCKBA-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DCKBA TO DCKBE-B-D

PRODUCT NAME: PDP11 45-11/40 BASIC CP TESTS

DATE CREATED: 15 NOV 1972

MAINTAINER: DIAGNOSTIC GROUP

AUTHORS: JOHN ADAMS

THIS GROUP OF TESTS CONSIST OF:

MAINDEC NO.	TEST FUNCTION	PROCESSOR
DCKBA-A	SXT INSTRUCTION	11/45,11/40
DCKBB-A	SOB INSTRUCTION	11/45,11/40
DCKBC-A	XOR INSTRUCTION	11/45,11/40
DCKBD-B	MARK INSTRUCTION	11/45,11/40
DCKBE-B	RTT INSTRUCTION	11/45,11/40

1.0 Abstract

This is the first 5 of 15 tests that incrementally test and isolate simple malfunctions in the PDP 11/45,11/40. The tests should be run in the indicated alphabetic sequence. There are additional tests for more complex malfunctions. All tests are executed in kernel mode only except for test DCKBE (11/45 only).

2.0 Requirements

2.1 Equipment

PDP11/45 or PDP 11/40

2.2 Storage

Program storage - The programs use all of a 4KW memory with the exception of 17500 to 17776 (which is reserved for the boot and absolute loader).

2.3 Preliminary Programs

Tests T0-T13(D0AA-D0MA)

3.0 Loading Procedure

Load program into memory using ABS loader.

4.0 Starting Procedure

Load address 200. Press start. The program will loop, and ring bell on completion. Pass count may be monitored in the display register (11/45 only), and is stored in address 1000.

5.0 Operating Procedure

5.1 Operational Switch Settings

SW<08>=1 Load PDP11/45 MICRO BREAK register with value in SW<00-07>. (At start of test only).

5.2 Subroutine Abstracts

5.2.1 SCOPE

SCOPE is a MOVE PC, R1(010701) and stores the PC+2 in R1 (or R11 if the register set bit is set in the PDP11/45 Processor Status Word.) Thus, R1 (R11) may be used as a tag to determine the last test sucessfully completed.

5.2.2 HLT

HLT is a HALT instruction and is executed whenever a hardware malfunction is detected by the diagnostic. The address lights display the PC-PC+2 of the halt instruction if an 11/40-11/45.

5.3 Program and/or Operator Action

5.3.1 PASS COUNT (ICNT)

The number of program passes completed is contained in address ICNT (1000). This address may be examined to determine in which pass the error occured. Note, the pass count is displayed in the display register in the PDP11/45.

6.0 Errors

6.1 Test Errors Will Cause a Halt

False trap/interrupt errors - the program will halt at the trap vector address +2.

6.2 Error Recovery

Test errors - press continue or loop test (see sec 6.3)

Trap errors - determine where error occurred (see sec 8)

6.3 Error Looping

To loop on an error replace the HLT instruction with a branch back to the previous SCOPE instruction. Note that if the error is intermittent that the test will drop through the HLT and proceed to the next test, therefore to loop the test continuosly replace the BEQ .+4 instruction immediatly preceeding the HLT with the branch back to the previous SCOPE.

To loop trap failures patch in the following routine at the address of the trap vector:

```
TRAPVEC:        TRAPVEC+4
TRAPVEC+2:      0
TRAPVEC+4       012716  ;move 'SCOPE'
                        ;address to stack
TRAPVEC+6:      address ;address of previous
                        ;SCOPE inst.
TRAPVEC+10:     000006  ;return to test at 'SCOPE'
```

Restore all locations before proceeding to next test(s).

7.0 Restrictions

These programs must be loaded in the lower 4K of memory.

7.1 Starting restriction

All programs must be initially started at 200 and may be started at a SCOPE instruction thereafter.

7.2 Operational restriction

None

8.0 Miscellaneous

If a halt occures in the trap or interrupt vector area, examine register 6 (the stack pointer). The contents of R6 contains the address where the PC of the instruction that caused the trap is stored.

8.1 Execution time

All tests take approximately 1 minute each on an 11/45 with core memory.

8.2 Stack pointer

All programs initially set the stack pointer at 500

9.0 Program description

DCKBA This is a test of the SXT instruction and insures correct results and condition code operation. The SXT instruction is tested in all address modes in a general register and the PC.

DCKBB This is a test of the SOB instruction and insures correct branching and condition code operation.

DCKBC This is a test of the XOR instruction and insures correct results and condition code operation. The XOR instruction is executed using various operands ard is executed using all address modes using a general register and the PC.

DCKBD This is a test of the MARK instruction. The test executes the MARK instruction using all values of 'N' and checks results. Correct condition code operation is also tested.

DCKBE This is a test of the RTT and RTI instructions and uses 'T' bit traps in the test. Proper stack operation is tested and also proper status changes are tested.

```
274                                     .TITLE  MAINDEC-11-DCKBA-B PDP11/45 SXT INST TEST
                                        .NLIST  MC,MD,SEQ
                                        .LIST   ME
                                        .ABS
                                ;TEST DCKBAA- TEST OF THE SXT INSTRUCTION AND INCLUDES DATA CHECKS AND
                                ;CONDITION CODES CHECKS.
```

```
                                ;       THE SXT INSTRUCTION EXTENDS THE SIGN (AS DETERMINED BY THE 'N'
                                ;BIT IN THE STATUS WORD) INTO THE DESTINATION ADDRESS, IE. IF 'N' IS
                                ;= TO 1 THEN THE DESTINATION RESULT WILL BE -1 (ALL 1'S), AND IF 'N' IS
                                ;= TO 0 THEN THE DESTINATION RESULT WILL BE 0 (ALL 0'S). THE CONDITION
                                ;CODES ARE AFFECTED AS FOLLOWS:
                                ;       'Z' IS SET IF THE RESULT IS 0,OTHERWISE 'Z' IS CLEARED
                                ;       'N' IS SET IF THE RESULT IS NEGATIVE,CLEARED IF POSITIVE
                                ;       'C' IS UNCHANGED
                                ;       'V' IS CLEARED

                                ;ITEMS NOT TESTED BY THIS TEST
                                ;       1: THAT SXT WILL SET ALL BITS IN THE PSW WHEN 1'S ARE EXTENDED (EX-
                                ;       CLUSIVE OF THE 'T' BIT)
                                ;STARTING PROCEEDURE
                                ;       LOAD ADDRESS =200
                                ;       PRESS START
                                ;       STACK POINTER IS SET AT 500
                                ;       BELL WILL RING WHEN TEST IS COMPLETE

                000000          HLT=HALT
                010701          SCOPE=010701                    ;MOV PC,R1
                177776          PSW=177776              ;ADDRESS OF PROCESSER STATUS WORD
                177770          UBREAK=177770           ;ADDRESS OF MICRO BREAK REGISTER
                177564          TPCSR=177564            ;ADDRESS OF TELEPRINTER CSR
                177566          TPBUF=177566            ;ADDRESS OF TELEPRINTER BUFFER
                177570          SWR=177570              ;ADDRESS OF CONSOLE SWITCH REGISTER
                177570          DISPLAY=177570          ;ADDRESS OF DISPLAY REGISTER
                                ;****************INITIAL STACK POINTER=0500**********************
                000500          STKPTR=0500                     ;INITIAL STACK POINTER
                000000                  .=0
      000000    000002                  .+2
      000002    000000                  HALT
      000004    000006                  .+2
      000006    000000                  HALT
      000010    000012                  .+2
      000012    000000                  HALT
      000014    000016                  .+2
      000016    000000                  HALT
      000020    000022                  .+2
      000022    000000                  HALT
      000024    000026                  .+2
      000026    000000                  HALT
      000030    000032                  .+2
      000032    000000                  HALT
      000034    000036                  .+2
      000036    000000                  HALT
      000040    000042                  .+2
      000042    000000                  HALT
      000044    000046                  .+2
      000046    000000                  HALT
      000050    000052                  .+2
      000052    000000                  HALT
      000054    000056                  .+2
      000056    000000                  HALT
      000060    000062                  .+2
      000062    000000                  HALT
```

```
000064  000066                  .+2
000066  000000                  HALT
000070  000072                  .+2
000072  000000                  HALT
000074  000076                  .+2
000076  000000                  HALT
000100  000102                  .+2
000102  000000                  HALT
000104  000106                  .+2
000106  000000                  HALT
000110  000112                  .+2
000112  000000                  HALT
000114  000116                  .+2
000116  000000                  HALT
000120  000122                  .+2
000122  000000                  HALT
000124  000126                  .+2
000126  000000                  HALT
000130  000132                  .+2
000132  000000                  HALT
000134  000136                  .+2
000136  000000                  HALT
000140  000142                  .+2
000142  000000                  HALT
000144  000146                  .+2
000146  000000                  HALT
000150  000152                  .+2
000152  000000                  HALT
000154  000156                  .+2
000156  000000                  HALT
000160  000162                  .+2
000162  000000                  HALT
000164  000166                  .+2
000166  000000                  HALT
000170  000172                  .+2
000172  000000                  HALT
000174  000176                  .+2
000176  000000                  HALT
000200  000202                  .+2
000202  000000                  HALT
000204  000206                  .+2
000206  000000                  HALT
000210  000212                  .+2
000212  000000                  HALT
000214  000216                  .+2
000216  000000                  HALT
000220  000222                  .+2
000222  000000                  HALT
000224  000226                  .+2
000226  000000                  HALT
000230  000232                  .+2
000232  000000                  HALT
000234  000236                  .+2
000236  000000                  HALT
000240  000242                  .+2
000242  000000                  HALT
```

```
000244  000246                  .+2
000246  000000                  HALT
000250  000252                  .+2
000252  000000                  HALT
000254  000256                  .+2
000256  000000                  HALT
000260  000262                  .+2
000262  000000                  HALT
000264  000266                  .+2
000266  000000                  HALT
000270  000272                  .+2
000272  000000                  HALT
000274  000276                  .+2
000276  000000                  HALT
000300  000302                  .+2
000302  000000                  HALT
000304  000306                  .+2
000306  000000                  HALT
000310  000312                  .+2
000312  000000                  HALT
000314  000316                  .+2
000316  000000                  HALT
000320  000322                  .+2
000322  000000                  HALT
000324  000326                  .+2
000326  000000                  HALT
000330  000332                  .+2
000332  000000                  HALT
000334  000336                  .+2
000336  000000                  HALT
000340  000342                  .+2
000342  000000                  HALT
000344  000346                  .+2
000346  000000                  HALT
000350  000352                  .+2
000352  000000                  HALT
000354  000356                  .+2
000356  000000                  HALT
000360  000362                  .+2
000362  000000                  HALT
000364  000366                  .+2
000366  000000                  HALT
000370  000372                  .+2
000372  000000                  HALT
000374  000376                  .+2
000376  000000                  HALT

        000000                  .=0
000000  000167  001634          JMP     SXT07

        000046                  .=46
000046  002604                  LOGICAL
        000052                  .=52
000052  040000                  40000
```

```
                000000                  DEST=%0
                000200                          .=200
000200  000167  000576                  START:  JMP     START
                001000                          .=1000
001000  000000                  ICNT:   0                       ;PASS COUNT
001002  005067  177772          START:  CLR     ICNT            ;CLEAR PASS COUNT
001006  012706  000500          BEGIN:  MOV     #STKPTR,%6      ;SET STACK POINTER
001012  016737  177762  177570        MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT
001020  032737  000400  177570        BIT     #400,@#SWR      ;LOAD MICRO BREAK REGISTER?
001026  001403                        BEQ     .+10            ;BRANCH IF NOT
001030  113737  177570  177770        MOVB    @#SWR,@#UBREAK  ;LOAD MICRO BREAK WITH SR0-7

                                ;TEST THAT WHEN N IS CLEAR SXT CLEARS THE DESTINATION
001036  010701                        SCOPE
001040  012700  177777                MOV     #-1,DEST
001044  000277                        SCC                     ;SET ALL CONDITION CODES
001046  000250                        CLN                     ;CLEAR N
001050  006700                        SXT     DEST            ;EXTEND N BIT (0) INTO DESTINATION
001052  005700                        TST     DEST            ;DESTINATION=0?
001054  001401                        BEQ     .+4
001056  000000                        HLT                     ;DESTINATION OTHER THAN 0

                                ;TEST THAT WHEN N IS SET THAT SXT SETS ALL BITS IN THE DESTINATION
001060  010701                        SCOPE
001062  005000                        CLR     DEST
001064  000257                        CCC                     ;CLEAR ALL CONDTION CODES
001066  000270                        SEN                     ;SET N
001070  006700                        SXT     DEST            ;EXTEND N BIT (1) INTO DESTINATION
001072  020027  177777                CMP     DEST,#-1        ;DID ALL BITS SET IN DESTINATION
001076  001401                        BEQ     .+4
001100  000000                        HLT                     ;DESTINATION OTHER THAN 177777

                                ;TEST THAT N IS CLEAR AFTER SXT EXTENDS 0'S INTO THE DESTINATION
001102  010701                        SCOPE
001104  012700  177777                MOV     #-1,DEST
001110  000250                        CLN
001112  006700                        SXT     DEST
001114  100001                        BPL     .+4
001116  000000                        HLT

                                ;TEST THAT N IS SET AFTER SXT EXTENDS 1'S INTO THE DESTINATION
001120  010701                        SCOPE
001122  005000                        CLR     DEST
001124  000270                        SEN
001126  006700                        SXT     DEST
001130  100401                        BMI     .+4
001132  000000                        HLT

                                ;TEST THAT SIGN EXTENDS PROPERLY AFTER A MOVE INSTRUCTION (N SET)
001134  010701                        SCOPE
001136  005000                        CLR     DEST
001140  012767  177777  177774        MOV     #-1,.+2
001146  006700                        SXT     DEST
001150  020027  177777                CMP     DEST,#-1
001154  001401                        BEQ     .+4
```

```
        001156  000000                  HLT
```

```
                                        ;TEST THAT SIGN EXTENDS PROPERLY AFTER A MOVE INSTRUCTION (N CLEAR)
001160  010701                                SCOPE
001162  012700  177777                       MOV     #-1,DEST
001166  012767  000000  177774               MOV     #0,.+2
001174  006700                               SXT     DEST
001176  020027  000000                       CMP     DEST,#0
001202  001401                               BEQ     .+4
001204  000000                               HLT

                                        ;TEST THAT SIGN EXTENDS PROPERLY AFTER VARIOUS BYTE INSTRUCTIONS
                                        ;NOTE: THESE TESTS MUST BE RUN SEQUENTIALLY OR TEMP & TEMP+2 MUST
                                        ;BE MANUALLY LOADED BEFORE STARTING THE TEST.THE COMMENT AT THE BE -
                                        ;INNING OF THE TEST SHOWS CONTENTS OF TEMP & TEMP+2.
                                                                ;(TEMP+1),(TEMP)/(TEMP+3),(TEMP+2)
001206  005000                               CLR     DEST       ;?,?/?,?
001210  005067  000226                       CLR     TEMP       ;0,0/?,?
001214  105167  000223                       COMB    TEMP+1     ;-1,0/?,?
001220  006700                               SXT     DEST
001222  022700  177777                       CMP     #-1,DEST
001226  001401                               BEQ     .+4
001230  000000                               HLT                ;ERROR! SIGN EXTEND FAILED (N=1)
001232  010701                               SCOPE

001234  005000                               CLR     DEST       ;-1,0/?,?
001236  105367  000200                       DECB    TEMP       ;-1,-1/?,?
001242  006700                               SXT     DEST
001244  022700  177777                       CMP     #-1,DEST
001250  001401                               BEQ     .+4
001252  000000                               HLT                ;ERROR! SIGN EXTEND FAILED (N=1)
001254  010701                               SCOPE

001256  005000                               CLR     DEST       ;-1,-1/?,?
001260  156767  000157  000156               BISB    TEMP+1,TEMP+2  ;-1,-1/?,-1
001266  006700                               SXT     DEST
001270  022700  177777                       CMP     #-1,DEST
001274  001401                               BEQ     .+4
001276  000000                               HLT                ;ERROR! SIGN EXTEND FAILED (N=1)
001300  010701                               SCOPE

001302  005000                               CLR     DEST       ;-1,-1/?,-1
001304  116767  000134  000133               MOVB    TEMP+2,TEMP+3  ;-1,-1/-1,-1
001312  006700                               SXT     DEST
001314  022700  177777                       CMP     #-1,DEST
001320  001401                               BEQ     .+4
001322  000000                               HLT                ;ERROR! SIGN EXTEND FAILED (N=1)
001324  010701                               SCOPE

001326  012700  177777                       MOV     #-1,DEST   ;-1,-1/-1,-1
001332  146767  000105  000105               BICB    TEMP+1,TEMP+3  ;-1,-1/0,-1
001340  006700                               SXT     DEST
001342  005700                               TST     DEST
001344  001401                               BEQ     .+4
001346  000000                               HLT                ;ERROR! SIGN EXTEND FAILED (N=0)
001350  010701                               SCOPE
```

```
001352  012700  177777                MOV     #-1,DEST        ;-1,-1/0,-1
001356  000261                        SEC                     ;SET CARRY
001360  105567  000057                ADCB    TEMP+1          ;0,-1/0,-1
001364  006700                        SXT     DEST
001366  005700                        TST     DEST
001370  001401                        BEQ     .+4
001372  000000                        HLT                     ;ERROR! SIGN EXTEND FAILED (N=0)
001374  010701                        SCOPE

001376  012700  177777                MOV     #-1,DEST        ;0,-1/0,-1
001402  105267  000034                INCB    TEMP            ;0,0/0,-1
001406  006700                        SXT     DEST
001410  005700                        TST     DEST
001412  001401                        BEQ     .+4
001414  000000                        HLT                     ;ERROR! SIGN EXTEND FAILED (N=0)
001416  010701                        SCOPE

001420  012700  177777                MOV     #-1,DEST        ;0,0/0,-1
001424  105167  000014                COMB    TEMP+2          ;0,0/0,0
001430  006700                        SXT     DEST
001432  005700                        TST     DEST
001434  001404                        BEQ     TZ
001436  000000                        HLT                     ;ERROR! SIGN EXTEN FAILED (N=0)
001440  010701                        SCOPE
001442  000000          TEMP:0
        001446                      .=.+2

                        ;TEST THAT Z BIT IS SET AFTER SXT EXTENDS 0'S
001446  010701          TZ:         SCOPE
001450  012700  177777              MOV     #-1,DEST
001454  000250                      CLN                     ;CLEAR N
001456  006700                      SXT     DEST            ;EXTEND N BIT (0) INTO DESTINATION
001460  001401                      BEQ     .+4             ;BRANCH IF Z IS SET
001462  000000                      HLT                     ;Z NOT SET AFTER 0'S WERE EXTENDED

                        ;TEST THAT Z BIT IS CLEAR AFTER SXT EXTENDS 1'S
001464  010701                      SCOPE
001466  005000                      CLR     DEST
001470  000270                      SEN
001472  006700                      SXT     DEST
001474  001001                      BNE     .+4
001476  000000                      HLT


                        ;TEST THAT THE CARRY BIT (C) IS UNCHANGED BY SXT (C=0,N=0)
001500  010701                      SCOPE
001502  000257                      CCC                     ;SLAER ALL CONDITION CODES
001504  006700                      SXT     DEST            ;EXTEND SIGN
001506  103001                      BCC     .+4             ;BRANCH IF CARRY IS CLEAR
001510  000000                      HLT                     ;ERROR  CARRY SET

                        ;TEST THAT THE CARRY BIT IS UNCHANGED BY SXT (C=0,N=1)
001512  010701                      SCOPE
001514  000257                      CCC
001516  000270                      SEN                     ;C=0,N=1
001520  006700                      SXT     DEST            ;EXTEND SIGN
```

```
001522  103001                  BCC     .+4             ;BRANCH IF CARRY IS CLEAR
001524  000000                  HLT                     ;DRROR! CARRY SET

                        ;TEST THAT THE CARRY BIT IS UNCHANGED BY SXT (C=1,N=0)
001526  010701                  SCOPE
001530  000257                  CCC
001532  000261                  SEC                     ;C=1,N=0
001534  006700                  SXT     DEST            ;EXTEND SIFN
001536  103401                  BCS     .+4             ;BRANCH IF CARRY IS SET
001540  000000                  HLT                     ;ERROR! CARRY CLEARED

                        ;TEST THAT CARRY IS UNCHANGED BY SXT (C=1,N=1)
001542  010701                  SCOPE
001544  000277                  SCC                     ;C=1,N=1
001546  006700                  SXT     DEST            ;EXTEND SIGN
001550  103401                  BCS     .+4             ;BRANCH IF CARRY SET
001552  000000                  HLT                     ;ERROR! CARRY CLEARED

                        ;TEST THAT THE V BIT IS CLEARED BY SXT (V=0,N=0)
001554  010701                  SCOPE
001556  000257                  CCC                     ;V=0,N=0
001560  006700                  SXT     DEST            ;EXTEND SIGN
001562  102001                  BVC     .+4             ;BRANCH IF V IS CLEAR
001564  000000                  HLT                     ;ERROR! V SET

                        ;TEST THAT V IS CLEARED BY SXT (V=0,N=1)
001566  010701                  SCOPE
001570  000257                  CCC
001572  000270                  SEN                     ;V=0,N=1
001574  006700                  SXT     DEST            ;EXTEND SIGN
001576  102001                  BVC     .+4             ;BRANCH ID V IS CLEAR
001600  000000                  HLT                     ;ERROR! V SET

                        ;TEST THAT V IS CLEARED BY SXT (V=1,N=0)
001602  010701                  SCOPE
001604  000277                  SCC
001606  000250                  CLN                     ;C=1,N=0
001610  006700                  SXT     DEST            ;EXTELD SIGN
001612  102001                  BVC     .+4             ;BRANNCH IF V IS CLEAR
001614  000000                  HLT                     ;ERROR! V SET
```

```
                                ;TEST THAT V IS CLEARED BY SXT (C=1,N=1)
001616  010701                          SCOPE
001620  000277                          SCC                     ;V=1,N=1
001622  006700                          SXT     DEST            ;EXTEND SIGN
001624  102001                          BVC     .+4     ;BRANCH IF V IS CLEAR
001626  000000                          HLT                     ;ERROR! V REMAINED SET

        000007                  DEST=%7
                                ;TEST THAT SIGN EXTENDS INTO R7 (N=0).
001630  010701                          SCOPE
001632  000257                          CCC                     ;N=0
001634  006707                          SXT     DEST            ;EXTEND 0'S INTO THE PC
001636  000000                          HLT                     ;ERROR! PC SHOULD'VE GONE TO 0
001640  000402                  SXT07:  BR      MODE67

        001642                  DEST=.
        001644                  .=.+2
001644  000000                  IDEST:  0
                                ;TEST DESTINATION MODE 67
001646  010701                  MODE67: SCOPE
001650  005067  177766                  CLR     DEST
001654  000270                          SEN                     ;SET N
001656  006767  177760                  SXT     DEST            ;EXTEND 1'S INTO DEST
001662  026727  177754  177777        CMP     DEST,#-1        ;DID 1'S EXTEND
001670  001401                          BEQ     .+4
001672  000000                          HLT                     ;1'S FAILED TO EXTEND

                                ;TEST DESTINATION MODE 27
001674  010701                          SCOPE
001676  005067  000004                  CLR     MODE27
001702  000270                          SEN                     ;SET N
001704  006727                          SXT     (7)+            ;EXTEND 1'S INTO NEXT LOCATION
001706  000000                  MODE27: 0
001710  026727  177772  177777        CMP     MODE27,#-1      ;DID 1'S EXTEND
001716  001401                          BEQ     .+4
001720  000000                          HLT

                                ;TEST DESTINATION MODE 37
001722  010701                          SCOPE
001724  012737  177777  001642        MOV     #-1,@#DEST
001732  000250                          CLN                     ;CLEAR N
001734  006737  001642                  SXT     @#DEST          ;EXTEND 0'S
001740  023727  001642  000000        CMP     @#DEST,#0       ;DID 0'S EXTEND
001746  001401                          BEQ     .+4
001750  000000                          HLT

                                ;TEST DESTINATION MODE 77
001752  010701                          SCOPE
001754  012767  001644  177660        MOV     #IDEST,DEST
001762  012777  177777  177652        MOV     #-1,@DEST
001770  000250                          CLN
001772  006777  177644                  SXT     @DEST
001776  027727  177640  000000        CMP     @DEST,#0
002004  001401                          BEQ     .+4
002006  000000                          HLT
```

```
                                ;TEST DESTINATION MODE 1
002010  010701                          SCOPE
002012  012700  002320                  MOV     #MODE1,%0
002016  005002                          CLR     %2
002020  000261                          SEC
002022  005602                          SBC     %2
002024  006710                          SXT     (0)
002026  022767  177777  000264          CMP     #-1,MODE1
002034  001401                          BEQ     .+4
002036  000000                          HLT                             ;ERROR! INCORRECT RESULT

                                ;TEST DESTINATION MODE2
002040  010701                          SCOPE
002042  005046                          CLR     -(6)                    ;SET UP STACK WITH 'NEW' STATUS
002044  012746  002056                  MOV     #MODE2,-(6.             ;AND 'NEW' PC
002050  012702  002320                  MOV     #MODE1,%2
002054  000002                          RTI
002056  006722                  MODE2:  SXT     (2)+
002060  005767  000234                  TST     MODE1
002064  001401                          BEQ     .+4
002066  000000                          HLT                             ;ERROR! INCORRECT RESULT
002070  022702  002322                  CMP     #MODE1+2,%2             ;CHECK AUTO-INCREMENT
002074  001401                          BEQ     .+4
002076  000000                          HLT                             ;ERROR! AUTO INCREMENT FAILED

                                ;TEST DESTINATION MODE 3
002100  010701                          SCOPE
002102  012767  002322  000210          MOV     #MODE3,MODE1
002110  012702  002320                  MOV     #MODE1,%2
002114  000270                          SEN
002116  006732                          SXT     @(2)+
002120  022767  177777  000174          CMP     #-1,MODE3
002126  001401                          BEQ     .+4
002130  000000                          HLT                             ;ERROR! INCORRECT RESULT

                                ;TEST DESTINATION MODE 4
002132  010701                          SCOPE
002134  012703  002322                  MOV     #MODE3,%3
002140  012767  177777  000152          MOV     #-1,MODE1
002146  005067  000150                  CLR     MODE3
002152  006743                          SXT     -(3)
002154  005767  000140                  TST     MODE1
002160  001401                          BEQ     .+4
002162  000000                          HLT                             ;ERROR! INCORRECT RESULT
002164  022703  002320                  CMP     #MODE3-2,%3             ;CHECK AUTO-DECREMENT
002170  001401                          BEQ     .+4
002172  000000                          HLT                             ;ERROR! AUTO-DECREMENT FAILED

                                ;TEST DESTINATION MODE 5
002174  010701                          SCOPE
002176  012767  002320  000116          MOV     #MODE1,MODE3
002204  012704  002324                  MOV     #MODE3+2,%4
002210  000270                          SEN
002212  006754                          SXT     @-(4)
002214  022767  177777  000076          CMP     #-1,MODE1
002222  001401                          BEQ     .+4
```

```
002224  000000                          HLT                     ;ERROR! INCORRECT RESULT
002226  022704  002322                  CMP     #MODE3,%4       ;CHECK AUTO-DECREMENT
002232  001401                          BEQ     .+4
002234  000000                          HLT                     ;ERROR! AUTO-DECREMENT DEFERRED FAILED

                                ;TEST DESTINATION MODE 6
002236  010701                          SCOPE
002240  012705  002322                  MOV     #MODE3,%5
002244  006765  177776                  SXT     -2(5)
002250  005767  000044                  TST     MODE1
002254  001401                          BEQ     .+4
002256  000000                          HLT                     ;ERROR! INCORRECT RESULT

                                ;TEST DESTINATION MODE 7
002260  010701                          SCOPE
002262  012767  177777  000030          MOV     #-1,MODE1
002270  012704  177770                  MOV     #-10,%4
002274  012767  002320  000020          MOV     #MODE1,MODE3
002302  006774  002332                  SXT     @MODE3+10(4)
002306  005767  000006                  TST     MODE1
002312  001401                          BEQ     .+4
002314  000000                          HLT                     ;ERROR INCORRECT RESULT
002316  000402                          BR      .+6             ;GO TO NEST TEST
002320  000000                  MODE1:  0
002322  000000                  MODE3:  0

                                ;TEST CONDITION CODES WHEN THE DESTINATION MODE IS NOT 0
                                ;N=0,Z=1,V=0,C=0
002324  010701                          SCOPE
002326  012767  177777  177764          MOV     #-1,MODE1
002334  000257                          CCC
002336  006767  177756                  SXT     MODE1
002342  016700  175430                  MOV     PSW,%0
002346  022700  000004                  CMP     #4,%0
002352  001401                          BEQ     .+4
002354  000000                          HLT                     ;ERROR! INCORRECT CONDITION CODES

                                ;N=0,Z=1,V=0,C=1
002356  010701                          SCOPE
002360  012767  177777  177732          MOV     #-1,MODE1
002366  012702  002320                  MOV     #MODE1,%2
002372  012767  000003  175376          MOV     #3,PSW
002400  006712                          SXT     (2)
002402  016700  175370                  MOV     PSW,%0
002406  022700  000005                  CMP     #5,%0
002412  001401                          BEQ     .+4
002414  000000                          HLT                     ;ERROR! INCORRECT CONDITION CODES

                                ;N=1,Z=0,V=0,C=0
002416  010701                          SCOPE
002420  012702  002320                  MOV     #MODE1,%2
002424  005067  177670                  CLR     MODE1
002430  012767  000016  175340          MOV     #16,PSW
002436  006722                          SXT     (2)+
002440  016700  175332                  MOV     PSW,%0
002444  022700  000010                  CMP     #10,%0
```

```
002450  001401                          BEQ     .+4
002452  000000                          HLT                     ;ERROR! INCORRECT CONDITION CODES

                                ;N=1,Z=0,V=0,C=1
002454  010701                          SCOPE
002456  012702  002322                  MOV     #MODE1+2,%2
002462  005067  177632                  CLR     MODE1
002466  012767  000017  175302          MOV     #17,PSW
002474  006742                          SXT     -(2)
002476  016700  175274                  MOV     PSW,%0
002502  022700  000011                  CMP     #11,%0
002506  001401                          BEQ     .+4
002510  000000                          HLT                     ;ERROR! INCORRECT CONDITION CODES


                                ;TEST THAT SXT EXTENDS 0'S INTO THE PSW
002512  010701                          SCOPE
002514  012767  000357  175254          MOV     #357,PSW
002522  000250                          CLN
002524  006767  175246                  SXT     PSW
002530  016700  175242                  MOV     PSW,%0          ;GET & TEST PSW CONTENTS
002534  001401                          BEQ     .+4
002536  000000                          HLT


002540  005267  176234                  INC     ICNT            ;INCREMENT PASS COUNT
002544  026727  176230  177777          CMP     ICNT,#-1
002552  001402                          BEQ     DONE            ;GO TO DONE IF 100. PASSES COMPLETED
002554  000167  176226                  JMP     BEGIN
002560  012767  000007  175000  DONE:   MOV     #7,TPBUF        ;RING BELL
002566  105767  174772                  TSTB    TPCSR           ;WAIT FOR THE
002572  100375                          BPL     .-4             ;BELL TO RING
002574  013702  000042                  MOV     @#42,%2         ;GET DECTAPE MONITOR ADDRESS
002600  001405                          BEQ     DONE1           ;DO NOT RETURN IF (42)=0
002602  000005                          RESET
002604  004712          LOGICAL:JSR     7,(2)           ;START OF
002606  000240                          NOP             ;ACT11
002610  000240                          NOP             ;OVERLAY
002612  000240                          NOP             ;AREA
002614  000167  176162          DONE1:  JMP     START           ;AND REPEAT TEST

        000001                          .END
```

```
BEGIN   001006          456#    855
DEST  = 001642          450#    464*    467*    468     474*    477*    478     484*    486*    492*    494*    500*    502*
                        503     509*    511*    512     521*    524*    525     530*    532*    535     538*    540*    541
                        546     548*    549     554*    556*    557     562*    565*    566     571*    573*    574     579*
                        581*    582     591*    593*    599*    601*    609*    617*    625*    632*    639*    647*    655*
                        661*    665*    669*    673*    678*    680*    681     697*    699*    700     706*    707*    709*
                        710
DISPLA= 177570          307#    457*
DONE    002560          854     856#
DONE1   002614          860     866#
HLT   = 000000          300#    470     480     488     496     505     514     527     535     543     551     559     568
                        576     584     595     603     611     619     627     634     641     649     657     663     670
                        683     693     702     712     723     734     737     747     757     760     770     773     781
                        791     805     816     827     838     849
ICNT    001000          454#    455*    457     852*    853
IDEST   001644          675#    706
LOGICA  002604          445     862#
MODE1   002320          716     721     729     732     735     741*    742     752*    755     764     768     779     785*
                        787     789     793#    799*    801*    809*    810     820     821*    831     832*
MODE2   002056          728     731#
MODE27  001706          687*    690#    691
MODE3   002322          741     745     751     753*    758     764*    765     771     777     787*    788*    794#
MODE67  001646          671     677#
PSW   = 177776          302#    802     811*    813     822*    824     833*    835     844*    846*    847
SCOPE = 010701          301#    463     473     483     491     499     508     528     536     544     552     560     569
                        577     585     590     598     607     614     622     630     637     644     652     659     667
                        677     686     696     705     715     726     740     750     763     776     784     798     808
                        819     830     843
START   001002          452     455#    866
STKPTR= 000500          309#    456
SWR   = 177570          306#    458     460
SXT07   001640          441     671#
TEMP    001442          522*    523*    531*    539*    547*    555*    564*    572*    580*    586#
TPBUF = 177566          305#    856*
TPCSR = 177564          304#    857
TZ      001446          583     590#
UBREAK= 177770          303#    460*
.     = 002620          310#    311     313     315     317     319     321     323     325     327     329     331     333
                        335     337     339     341     343     345     347     349     351     353     355     357     359
                        361     363     365     367     369     371     373     375     377     379     381     383     385
                        387     389     391     393     395     397     399     401     403     405     407     409     411
                        413     415     417     419     421     423     425     427     429     431     433     435     437
                        440#    444#    446#    451#    453#    459     469     479     487     495     501*    504     510*
                        513     526     534     542     550     558     567     575     587#    594     602     610     618
                        626     633     640     648     656     662     673     674#    682     692     701     711     722
                        733     736     746     756     759     769     772     780     790     792     804     815     826
                        837     848     858
```

```
ADCB    564
BCC     610  618
BCS     626  633
BEQ     459  469  479  504  513  526  534  542  550  558  567  575  583  594  682
        692  701  711  722  733  736  746  756  759  769  772  780  790  804  815
        826  837  848  854  860
BICB    555
BISB    539
BIT     458
BMI     495
BNE     602
BPL     487  858
BR      671  792
BVC     640  648  656  662
CCC     475  608  615  523  638  645  668  800
CLN     466  485  592  654  698  708  845
CLR     455  474  492  500  521  522  530  536  546  599  678  687  717  727  753
        821  832
CMP     478  503  512  525  533  541  549  681  691  700  710  721  735  745  758
        768  771  803  814  825  836  853
COMB    523  580
DECB    531
HALT    300  312  314  316  318  320  322  324  326  328  330  332  334  336  338
        340  342  344  346  348  350  352  354  356  358  360  362  364  366  368
        370  372  374  376  378  380  382  384  386  388  390  392  394  396  398
        400  402  404  406  408  410  412  414  416  418  420  422  424  426  428
        430  432  434  436  438
INC     852
INCB    572
JMP     441  452  855  866
JSR     862
MOV     456  457  464  484  501  509  510  554  562  571  579  591  697  706  707
        716  728  729  741  742  751  752  764  765  777  785  786  787  799  802
        809  810  811  813  820  822  824  831  833  835  844  847  856  859
MOVB    460  547
NOP     863  864  865
RESET   861
RTI     730
SBC     719
SCC     465  631  653  660
SEC     563  624  718
SEN     476  493  600  616  646  679  688  743  766
SXT     467  477  486  494  502  511  524  532  540  548  556  565  573  581  593
        601  609  617  625  632  639  647  655  661  669  680  689  699  709  720
        731  744  754  767  778  788  801  812  823  834  846
TST     468  557  566  574  582  732  755  779  789
TSTB    857
.ABS    277
.END    868
.LIST   276
.NLIST  275
.REM      1
.REPT   311
.TITLE  274
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DCKBAB/SOL/CRF/PAGNUM=DCKBAB
RUN-TIME: 1 3 .5 SECONDS
RUN-TIME RATIO: 58/6=8.6
CORE USED: 6K (11 PAGES)

Spooler runtime 2 Secpds, 16 KCS, 54 disk reads, 0 disk writes, 20 pages